MMF 9000

Systemüberblick



commodore
COMPUTER

MMF 9000

Systemüberblick

Urheberrechtliche Informationen

Die in diesem Dokument enthaltenen Informationen sind Eigentum der Waterloo Computing Systems Ltd. und dürfen ohne vorherige schriftliche Genehmigung eines dazu autorisierten Repräsentanten des Eigentümers weder im Ganzen noch in Teilen veröffenlicht, vervielfältigt oder anderweitig abgespeichert werden.

Waterloo Computing Systems Ltd. gibt keinerlei Garantie für die Richtigkeit und Vollständigkeit des Dokumentinhaltes oder der hier beschriebenen Programme oder für das Erreichen bestimmter Ergebnisse. Unter keinen Umständen können Waterloo Computing Systems Ltd., seine Mitarbeiter, Lieferanten, Händler oder Verfasser dieser Dokumentation mittelbar oder unmittelbar für eventuelle Schäden, Verluste, Kosten, Forderungen, Honorare oder Kosten jeglicher Art verantwortlich gemacht werden.

Commodore GmbH
Postfach 71 01 26
D-6000 Frankfurt 71

Commodore AG
Aeschenvorstadt 57
CH-4010 Basel

Steiner Computer-
Vertriebs GmbH
Fleschgasse 2
A-1130 Wien

INHALTSVERZEICHNIS

SEITE

INHALTSVERZEICHNIS

SEITE

INHALTSVERZEICHNIS

SEITE

INHALTSVERZEICHNIS

SEITE

INHALTSVERZEICHNIS

WATERLOO COMPUTING KUNDENINFORMATIONS-SERVICE

Die in diesem Dokument beschriebene Software wurde von Waterloo Computing System Ltd. implementiert. Es ist beabsichtigt, auch dieses System von Zeit zu Zeit zu verbessern und zu erweitern und neue Systeme anzubieten.

Die Kundeninformationsblätter werden in regelmäßigen Abständen veröffentlicht, um die Systembenutzer über die zuletzt stattgefundenen Entwicklungen hinsichtlich der Waterloo-Software zu unterrichten. Aus diesem Grund stellen diese Kundeninformationsblätter den direktesten Weg dar, den Benutzter mit den neuesten Entwicklungen zu versorgen. Weitere Informationen im Zusammenhang mit der Veröffentlichung von Systementwicklungen werden bei Benachrichtigung der folgenden Adresse zur Verfügung gestellt:

Waterloo Computing Systems Newsletter
Box 943,
Waterloo, Ontario,
Canada
N2J 4C3

KAPITEL 1

EINLEITUNG

Dieses Dokument wurde geschaffen, um eine Einführung in und einen Überblick über den Commodore MMF 9000 zu geben. Dieses Ziel wird in den nächsten zwei Kapiteln angestrebt, die einen Überblick über die Hardware und die Waterloo MicroSoftware-Produkte bieten. In diesem Handbuch befinden sich auch einige Kapitel mit Themen, die sich auf mehrere Software-Produkte beziehen. Dabei befassen sich drei Kapitel mit dem Dateiverwaltungssystem, welches von der Waterloo MicroSoftware benutzt wird. Ein weiteres Kapitel behandelt den Bildschirm-Editor, der für die Eingabe und Bearbeitung von Texten und Programmen eingesetzt werden kann. Zusätzlich befindet sich in diesem Dokument ein Kapitel über das 'Setup'-Programm, welches bei der Inbetriebnahme der seriell betriebenen Datenübertragung (RS232) behilflich ist.

In den Anhängen befindet sich ein Überblick über die Waterloo-Bibliothek, eine Beschreibung der Gleitkomma-Emulation, die in Verbindung mit der Waterloo MicroSoftware zu benutzen ist und eine Beschreibung der Benutzung des Bildschirms unter Verwendung des Computersystems.

Einem noch nicht mit dem System vertrauten Benutzer wird empfohlen, die Kapitel 1 bis 3 eingehend zu studieren, wonach die Programmiererhandbücher der entsprechenden Programmiersprache zu lesen sind. Zu einem späteren Zeitpunkt können die anderen Teile dieses Dokumentes eingesehen werden, abhängig von den individuellen Anforderungen des Benutzers.

KAPITEL 2

COMMODORE MMF 9000-HARDWARE

2.1 EINLEITUNG

Dieses Kapitel dient als Einleitung für die Hardware des Commodore MMF 9000. Der erste Teil dieses Kapitels befaßt sich mit den Systemkomponenten und der zweite Teil mit Informationen zur internen Hardware-Architektur. Die Fähigkeiten des Systems können bereits überschaut werden, wenn nur der erste Teil dieses Kapitels eingesehen wird. Der zweite Teil dieses Kapitels wurde für diejenigen Benutzer geschrieben, die zusätzliche Informationen benötigen, um die Hardware und die technischen Aspekte besser verstehen zu können.

2.2 SYSTEMKOMPONENTEN - ÜBERBLICK

Der Commodore MMF 9000 besteht aus einer Tastatur, einem Bilschirm, zwei Mikroprozessoren und einer Anzahl von Anschlüssen, mit denen Peripheriegeräte verbunden werden können.

2.2.1 Mikroprozessor-Auswahl

Über einen 3-stufigen Kippschalter wird der gewünschte Mikroprozessor gewählt:

a) In Position '6502' wird der MOS-6502-Mikroprozessor angesprochen;
b) In der Position '6809' wird der Motorola MC6809 Mikroprozessor angesprochen. Soll die Waterloo MikroSoftware eingesetzt werden, dann ist diese Schalterstellung zu wählen;
c) In der Position 'PROG' wird der angesprochene Mikroprozessor durch die sich im Computer befindende Software selektiert.

In diesem Computer sind daher zwei Systeme in einem vereint.

2.2.2 Hauptspeicher

Der Hauptspeicher des Systems besteht aus einem Read/Write-Memory (RAM), in welchem Programme und Daten geladen und/oder ausgeführt werden können. Jeder Mikroprozessor verfügt darüber hinaus über einen individuellen Read-Only-Speicher (ROM), der Informationen enthält, die nicht geändert werden können. Bei dem 6502-Prozessor enthält dieser ROM die Software Commodore BASIC, die in dem betreffenden Handbuch beschrieben ist. Bei dem 6809-Mikroprozessor enthält dieser ROM eine Bibliothek nützlicher Routinen, die von der Waterloo MikroSoftware benutzt werden.

Der Read/Write-Speicher bietet eine Kapazität von 32.768 Bytes (32KB) Standardspeicher, auf den das Computerprogramm ständig zugreifen kann. Zusätzlich steht eine RAM-Speicherkapazität von 65.536 Bytes (64KB) zur Verfügung, die in 16 Bereiche zu je 4.096 Bytes (4KB) organisiert ist. Zu einem beliebigen Zeitpunkt ist nur einer dieser Bereiche der Software verfügbar. Es besteht eine Einrichtung, die es erlaubt, den zugriffsbereiten Bereich durch ein Programm austauschen zu lassen. In der Waterloo MikroSoftware wird diese Einrichtung sehr intensiv genutzt. Software-Programme, wie z.B. Interpreter, werden in diese austauschbaren Bereiche geladen, wodurch die Standard-Hauptspeicherkapazität von 32KB für das Anwenderprogramm (in einer der Mikro-Sprachen geschrieben) und die dazugehörigen Daten reserviert bleibt.

Ist ein Interpreter einmal geladen, dann modifizieren die Waterloo MikroSoftware-Interpreter diese austauschbaren Bereiche nicht. Die Befehle des Interpreters können jetzt vor nicht beabsichtigter Modifikation geschützt werden, indem der für die Hauptspeicherauswahl bestimmte 3-stufige Kippschalter auf 'READ' gesetzt wird. Dadurch wird verhindert, daß die austauschbaren RAM-Bereiche verändert werden. Die

Schalterstellung auf 'R/W' erlaubt die Modifikation des Speicherinhalts, und die Setzung des Schalters auf 'PROG' schafft die Voraussetzungen, daß der Hauptspeicherschutz durch die Software bestimmt wird.

2.2.3 Tastatur

Die Tastatur des MMF 9000 besteht aus 73 Tasten, die entsprechend einer gewöhnlichen Schreibmaschinentastatur ausgelegt sind, wobei Groß- und Kleinschreibung, numerischer Tastenblock und umfassende Cursor-Steuerungen zur Verfügung stehen. Die Betätigung einer dieser Tasten kann von der Software erkannt werden, die in der Anlage derzeitig ausgeführt wird. Normalerweise wird ein eingegebenes Zeichen auch auf dem Bildschirm angezeigt.

Aus diesem Grund ist es möglich, daß unterschiedliche Software-Programme die einzelnen Tasten unterschiedlich interpretieren. Das Waterloo-System MicroAPL kann einer Taste einen APL-Operator zuordnen. Commodore BASIC kann das gleiche Zeichen als ein graphisches Zeichen ansehen.

2.2.4 Bildschirm

Die auf dem Bildschirm angezeigten Zeichen sind in 25 Zeilen zu je 80 Zeichen angeordnet. Jedes Zeichen wird durch eine 8 x 8 Punktematrix auf den Bildschirm dargestellt, wobei normalerweise für ein Zeichen nur 7 x 7 Punkte benutzt werden. Unter normalen Umständen zeigt die Hardware die Zeilen mit zwei zusätzlichen gelöschten Punktereihen an, welche die eine Zeile von der anderen Zeile trennen. Diese Trennungsreihen können ausgelassen werden (durch Software-Steuerung; siehe Anhang über Bildschirmbenutzung), womit die Darstellung graphischer Zeichen verbessert wird.

Für den Bildschirm stehen 4 unterschiedliche Zeichensätze zur Verfügung. Ein Zeichensatz ist eine Ansammlung von alphanumerischen Zeichen und Sonderzeichen, die auf dem Bildschirm angezeigt werden. Diese Zeichensätze sind wie folgt:

1. Commodore graphischer Zeichensatz,
2. Commodore Groß-/Kleinbuchstaben Roman-Zeichensatz,
3. Waterloo Groß-/Kleinbuchstaben Roman-Zeichensatz,
4. Waterloo APL-Zeichensatz.

Die Zeichensätze werden im Anhang C näher beschrieben. Jedes Software-Produkt wählt den für den speziellen Zweck benötigten Zeichensatz. Das bedeutet, daß Waterloo MicroAPL den Zeichensatz Nr. 4 und Commodore BASIC die Zeichensätze 1 und 2 ansteuern. Ein Zeichensatz kann auch ausgehend von einem Anwenderprogramm, welches in MicroFORTRAN geschrieben ist, selektiert werden, was bedeutet, daß der Commodore Graphikzeichensatz zur Ausgabe graphischer Darstellungen eingesetzt werden kann.

2.2.5 Anschlüsse

Die Anlage verfügt über eine Reihe von Eingabe/Ausgabe-Anschlüssen, mit denen verschiedene Peripheriegeräte verbunden werden können. Die RS232-Anschlüsse können benutzt werden, um asynchron-seriell betriebene Geräte unter Verwendung dieser Schnittstelle einsetzen zu können. Das SETUP-Programm (im folgenden Kapitel beschrieben) erlaubt die Benutzung dieses Anschlusses mit unterschiedlichen Merkmalen, wie z.B. verschiedenen Datenübertragungsgeschwindigkeiten. Der RS232-Anschluß wird üblicherweise dazu benutzt, eine Verbindung zum Zentralrechner oder zu einem seriell betriebenen Drucker herzustellen.

Darüber hinaus besteht ein Anschluß mit der Schnittstelle IEEE-488, der für die Verbindung zu den Commodore Diskettenlaufwerken 2040, 4040 und 8050 und Druckern, Modell 2022 und 4022, vorgesehen ist. Eine Reihe dieser Geräte könne parallel geschaltet werden, was bedeutet, daß gleichzeitig mehr als ein Gerät dem System zur Verfügung steht.

Ferner ist eine parallel betriebene Schnittstelle von 8 Bits verfügbar. Diese wird in dem entsprechenden Commodore Handbuch beschrieben.

Die zwei vorhandenen Kassettenanschlüsse können im Zusammenhang mit dem Microcomputer-System 6502 eingesetzt werden, jedoch derzeitig noch nicht mit der 6809-Software. Diese Kassettenanschlüsse ermöglichen den Einsatz von Kassettengeräten, womit der Benutzer in die Lage versetzt wird, Dateien auf Tonbandkassetten zu speichern.

2.3 SYSTEMINTERNE INFORMATIONEN

Dieser Abschnitt befaßt sich mit der Einführung in die systeminternen Hardware-Aspekte des Commodore MMF 9000. Andere Dokumente sollten für eine detailliertere Beschreibung der einzelnen Komponenten hinzugezogen werden. Es wird in diesem Teil des Kapitels angenommen, daß der Leser über einige Hardware-Kenntnisse verfügt.

2.3.1 Hauptspeicherbelegung

Beide Mikrocomputer-Systeme betrachten den Hauptspeicher als einen Adreßbereich von 64KB, in welchem die folgenden Adressenunterbereiche (hexadezimal) zur Verfügung stehen:

Bandbreite	**Inhalt**
0000-7FFF	RAM
8000-8FFF	Bildschirmspeicher
9000-9FFF	ein Bereich von 'austauschbarem' RAM von 4KB
A000-E7FF	prozessorabhängiger ROM
E800-EFFF	Steuerregister (E/A-Adressen)
F000-FFFF	prozessorabhängiger ROM

Wird von einem Prozessor auf einen anderen umgeschaltet (entweder manuell durch Betätigung des Kippschalters oder durch Programmsteuerung) treten folgende Ereignisse ein:

1. der Speicher wird neu aufgeteilt, so daß die ROM-Adressen entsprechend der Struktur des angewählten Prozessors verfügbar sind;
2. der Prozessor beginnt mit der Ausführung. Der 6809-Prozessor fängt mit der Ausführung bei Speicheradresse FFFE an und der 6502 bei der Speicheradresse FFFC.

2.3.2 Speicherbereichaustausch

Nur einer der 16 verfügbaren austauschbaren Speicherbereiche von 4KB befindet sich im Hauptspeicher innerhalb des Adreßbereichs 9000-9FFF. Der Austausch des Speicherbereichs erfolgt durch Übergabe der neuen Speicherbereichnummer (0-F) an das Speicherbereich-Selektionsregister (siehe Beschreibung Abschnitt 2.3.4).

2.3.3 Adressen im E/A-Seite

Die folgenden Geräte- oder Steuerregister sind in der Eingabe/Ausgabe-Seite gefunden enthalten:

Adresse	Register
E810-E813	PIA
E820-E823	PIA
E840-E84F	VIA
E880-E881	Bildschirm-Steuereinheit
EFF0-EFF3	ACIA (6551)
EFF4-EFF5	ACIA (6850)
EFF8*	System-Latch
EFFC	Speicherbereich-Latch

* benötigt auch den Speicherbereich-Latch (Bit 7 = 1)

2.3.4 Speicherbereich-Latch

Dieses Register wird für die Selektion des austauschbaren Speicherbereichs benutzt, der zu dem jeweiligen Zeitpunkt in den Speicheradressen 9000 - 9FFF zugriffsbereit sein muß. Das Register ist ein 1-Byte-Latch.

Bit	Zweck, Bedeutung
0-3	Speicherbereichnummer
4-6	nicht benutzt
7	= 1, wenn System-Latch aktiv; anderweitig = 0

Das bedeutet, daß der hexadezimale Wert 87 (binär = 10000111) dazu führt, daß der Speicherbereich 7 zugriffsbereit ist.

2.3.5 System-Latch (EFF8)

Diese Adresse wird für die Auswahl (unter Programmsteuerung) des Mikroprozessor-Systems und des Speicherschutzes für die austauschbaren Speicherbereiche benutzt.

Bit	Zweck, Bedeutung
0	CPU-Auswahl (1 = 6502, 0 = 6809)
1	Speicherschutz (1 = Lesen/Schreiben, 0 = nur Lesen)
2	nicht benutzt
3	Fehlerdiagnose
4-7	nicht benutzt

Diese Bit-Muster haben nur dann Bedeutung, wenn der Kippschalter wie vorher beschrieben in Stellung 'PROG' gebracht wurde. Das Bit 3 sollte auf '1' gesetzt werden, bevor auf den 6502-Prozessor umgeschaltet wird. Dadurch wird verhindert, daß die Standard-Software des Prozessors den Inhalt des RAMSpeichers überschreibt und daher zerstört.

KAPITEL 3

BESCHREIBUNG DER WATERLOO MICROSOFTWARE

3.1 ÜBERSICHT ÜBER DIE WATERLOO MICROSOFTWARE

Es wurde eine umfassende Palette verschiedener Software-Produkte entwickelt, die den unterschiedlichsten Schulungsanforderungen der Universität von Waterloo gerecht wird. Diese auf andere Systeme übertragbare Software ist besonders für den Einsatz im Zusammenhang mit Mikrocomputern geeignet. Zu einem späteren Zeitpunkt wird jedoch in ihrer Funktion identische Software geliefert, die auf mittleren und großen Computeranlagen laufen kann. Das bedeutet, daß der Benutzer in Zukunft nicht mehr auf die Kapazitäten der Mikroanlagen begrenzt ist. Diese Programme werden in der Lage sein, ohne Modifikationen auf großen und schnelleren Computersystemen zu laufen.

Die Software-Palette besteht aus Interpretierern für die verschiedenen Programmiersprachen, aus einem Editor, einem Betriebssystem und einem auf ASSEMBLER basierenden Programmentwicklungssystem.

3.1.1 Programmiersprachen-Interpretierer

Folgende Programmiersprachen-Interpretierer sind vorhanden:

Waterloo MicroBASIC,
Waterloo MicroPASCAL,
Waterloo MicroFORTRAN,
Waterloo MicroAPL.

Diese Programmiersprachen-Interpreter wurden speziell für Schulungszwecke für das Abhalten von Programmierungsseminaren entwickelt. Das Konzept der Interpreter beinhaltet hervorragende Möglichkeiten der Fehlerdiagnose und -korrektur, die besonders im Zusammenhang mit der Vermittlung von Programmierungsmethoden im Schulungsalltag und bei der Programmentwicklung behilflich sind.

3.1.1.1 Waterloo MicroBASIC

Waterloo MicroBASIC beinhaltet ANS Minimal BASIC, mit einer geringfügigen Ausnahme und verschiedenen Erweiterungen, wie z.B. die strukturierte Programmsteuerung, langen Namen für Variablen und andere Programmelemente, Zeichenketten, Manipulation, aufrufbare Prozeduren und Multi-Line-Funktionen, sequentielle und relative Dateiadressierung, Integer-Arithmetik, Korrektureinrichtungen und benutzerfreundliche Methoden für die Programmeingabe und Korrektur von Programmen.

3.1.1.2 Waterloo MicroPASCAL

Bei Waterloo MicroPASCAL handelt es sich um die umfassende Implementierung von PASCAL. Dieses Produkt entspricht der derzeitig zur Verfügung stehenden Entwurfsversion, die von dem PASCAL-Kommittee der International Standard Organization (ISO) vorbereitet wird. Die ISO-Empfehlungen sind eine Verfeinerung der Sprache, die ursprünglich von Wirth definiert wurde und weichen nur in wenigen Details davon ab. Waterloo MicroPASCAL enthält sehr fortschrittliche Einrichtungen, wie z.B. Textdateiunterstützung, Referenzvariablen und Tabellen mit mehreren Dimensionen. Darüber hinaus verfügt das Produkt über eine sehr leistungsfähige Einrichtung zur interaktiven Fehlerbeseitigung.

3.1.1.3 Waterloo MicroFORTRAN

Hierbei handelt es sich um eine FORTRAN-Sonderversion, die besonders für Schulungszwecke entwickelt wurde. Viele der im normalen FORTRAN enthaltenen Merkmale sind im Waterloo MicroFORTRAN vorhanden, wobei es jedoch erheblich von der üblichen Standards dieser Sprache abweicht. Dieser Programmiersprachen-Prozessor verfügt über sehr wesentliche Merkmale des Compilers WATFIV-S, welcher häufig bei IBM-Anlagen Verwendung findet, sowie einige zusätzliche Einrichtungen aus den Definitionen FORTRAN-77. So z.B. werden die folgenden Funktionen unterstützt: FORMAT, Unterroutinen und Funktionen, Mehrdimensionaltabellen, erweiterte Zeichensatzmanipulation, strukturierte Programmiersteuerung und Datei-Eingabe/Ausgabe. Zusätzlich stellt der Interpretierer sehr leistungsfähige interaktive Korrekturmöglichkeiten zur Verfügung.

3.1.1.4 Waterloo MicroAPL

Hinsichtlich der Syntax und des Aufbaus von APL-Anweisungen, der Operatoren und einfachen Funktionen, Eingabe/Ausgabe-Formen und definierten Funktionen wurde Waterloo MicroAPL als vollständige und umfassende Implementierung der IBM/ACM-Standards für APL entwickelt. Systembefehle, Systemvariablen und Systemfunktionen entsprechen denen einer Einbenutzer-Umgebung. Größe und Aufbau der Tabellen sowie die Längen der Namen sind nur wenigen Einschränkungen unterworfen. Der gemeinsam benutzte variable Prozessor wurde nicht implementiert. Die Erweiterungen beinhalten Systemfunktionen, die Dateien im Zusammenhang mit APL-Tabelle unterstützen. Darüber hinaus ist das APL-Äquivalent der BASIC-Befehle PEEK, POKE und SYS verfügbar.

3.1.2 Waterloo MicroEDITOR

Dieses Produkt eignet sich besonders für die Erstellung und Bearbeitung von Programmen und Dateien. Es handelt sich hierbei um einen nach herkömmlichen Methoden aufgebauten, zeilenorientierten Texteditor mit leistungsfähigen Textsuch- und Textaustausch-Befehlen, einschließlich der Möglichkeit zur Änderung der gesamten Datei. Die Unterstützung des gesamten Bildschirms und die Fähigkeit der Verwendung der Sonderfunktionstasten ermöglicht die Änderung, Hinzufügung und Löschung von Zeichen auf dem Bildschirm, ohne Befehle einsetzen zu müssen. Darüber hinaus besteht die Möglichkeit, vorher abgesetzte Befehle zu wiederholen und zu ändern, wodurch die Benutzerfreundlichkeit und Leistungsfähigkeit des Editors noch erweitert wird.

3.1.3 Dateiorientierter ASSEMBLER und Linkage-Editor

Der als Waterloo 6809 ASSEMBLER und LINKER bekannte ASSEMBLER und Linkage-Editor sind enthalten. Sie unterstützen die Entwicklung von allgemein gültigen, maschinensprachen-orientierten Programmen für den Prozessor Motorola 6809. Der ASSEMBLER unterstützt die Syntax und Richtlinien der ASSEMBLER-Sprache des Motorola 6809 und enthält sehr leistungsfähige Makro-Funktionen. Zusätzlich unterstützt der ASSEMBLER Pseudo-Befehle für die strukturierte Programmierung, lange Namen für die aussagekräftige Identifikation von Programmsegmenten und Datenfeldern und die Möglichkeit, Definitionen von separaten Dateien zu übernehmen. Der ASSEMBLER erstellt sowohl eine Liste als auch eine verschiebbare Obektdatei.

Der LINKER ermöglicht die Zusammenfügung einer beliebigen Anzahl verschiebbarer Objektdateien zur Erstellung einer ladbaren und ausführbaren Programmdatei. Da der LINKER diskettenorientiert ist, ist er in der Lage, ausführbare Programme zu erstellen, die größer als der zur Verfügung stehende RAM-Arbeitsplatz sind. Der LINKER unterstützt den Aufbau von Programmen in Segmenten oder austauschbaren Bereichen für die Ausführung im Zusammenhang mit dem austauschbaren Bereich des RAM-Hauptspeichers sowie den Aufbau von Programmen zu Ausführung innerhalb des normalen RAM-Hauptspeichers.

3.1.4 Waterloo MicroSUPERVISOR

Hierbei handelt es sich um ein Betriebssystem, welches für den Einsatz als persönlicher Computer eines Benutzers konzipiert wurde. Das Betriebssystem beinhaltet einen Monitor, eine Bibliothek und die seriell betriebene Datenübertragung, wie in den folgenden Absätzen beschrieben.

Der Monitor unterstützt das Laden von LINKER-erstellten Programmdateien in die austauschbaren RAM-Bereiche oder in den normalen RAM-Hauptspeicher. Der Monitor stellt auch einige Funktionen zur Verfügung, die bei der Fehlerbeseitigung von maschinesprache-orientierten Programmen behilflich sind. Unter diesen Funktionen befinden sich Befehle, mit den Hauptspeicheradressen und Mikroprozessor-Register des Prozessors 6809 angezeigt und geändert werden können, wobei für den besseren Überblick die Full-Screen-Einrichtung benutzt wird. Zusätzlich ist einer der Monitor-Befehle in der Lage, die Mikroprozessor-Instruktionen des Motorola 6809 zu demontieren und in mnemonische Form der ASSEMBLER-Sprache umzuwandeln.

Ferner ist in dem Betriebssystem eine aus Funktionen und Prozeduren bestehende Bibliothek enthalten, die von anderen Programmen dieser Produktpalette zur Ausführung herangezogen werden. Die Bibliothek unterstützt Funktionen für die Eingabe/Ausgabe-Operationen im Zusammenhang mit der Tastatur, dem Bildschirm und den Peripheriegeräten. Andere Teile der Bibliothek stellen Gleitkomma-Arithmetik, trigonometrische Funktionen und allgemeine Dienstleistungsfunktionen zur Verfügung.

Außerdem ist in dem Betriebssystem ein Programm zur Unterstützung seriell betriebener Datenkommunikationsleitungen enthalten, welches die Auswahl von programmierbaren Merkmalen erlaubt, wie z.B. die Datenübertragungsrate (Baud-Rate) der seriell betriebenen RS-232-Leitungen. Zusätzlich ist dieses Programm in der Lage, die Kommunikation zum Zentralrechner unter Verwendung einer seriell betriebenen Leitung aufzubauen, so daß auf die dort vorhandenen Dateien und Peripheriegeräte zugegriffen werden kann.

3.2 INBETRIEBNAHME DER WATERLOO-SOFTWARE

Nach Einschalten des MMF 9000 sind folgende Operationen auszuführen:

- der Hauptspeicher-Schreibschutzschalter ist auf R/W zu stellen;
- der Prozessorschalter ist auf 6809 zu stellen;
- danach ist der MMF 9000 einzuschalten (Schalter an der Rückseite);
- die Waterloo-Diskette ist in das linke Laufwerk einzuführen.

Daraufhin erscheint auf dem Bildschirm das Hauptmenü.

Eine Komponente der Waterloo Software kann ausgewählt werden, indem der abgekürzte Name der Komponente (wie im Menü angezeigt) eingegeben wird. So z.B. wird Waterloo MicroFORTRAN durch die Betätigung der Taste 'f' und danach der RETURN-Taste geladen.

Andere 6809-Programme können auch geladen und für die Ausführung bereitgestellt werden, indem der Name der Datei, die das Programm enthält, eingegeben wird. Ist bei dem Dateinamen keine Gerätespezifikation zu finden, dann versucht das System die entsprechende Datei auf der Diskette zu finden, die sich im linken Diskettenlaufwerk befindet. Das Format des Dateinamens wird in einem der folgenden Kapitel dieses Handbuches beschrieben. Für weitere Details über die Erstellung von 6809-Programmen ist das Handbuch WATERLOO 6809 ASSEMBLER einzusehen.

KAPITEL 4

DATEIINFORMATIONEN - ALLGEMEIN

4.1 EINLEITUNG

Diese Kapitel enthält Informationen über Dateien und Geräte, die dazu verwendet werden, Daten zu speichern, von dort wieder zu lesen und auf Bildschirm anzuzeigen. So z.B. kann eine Diskette dazu benutzt werden, eine Datei der Benotung von Schülern zu speichern. Es ist möglich, auf diese Datei durch andere Programme zuzugreifen, um z.B. Auswertungen oder Zusammenfassungen über die Benotungen zu erstellen. Ein Gerät, wie z.B. ein Drucker kann auch als Datei angesehen werden, obwohl es nicht möglich ist, Daten von dieser Datei zu lesen. Wird es trotzdem versucht, dann kommt es zu einem Fehler des Systems.

Erfolgt die Programmierung in einer der Waterloo Programmiersprachen, dann wird der bestimmte Dateiname oder das Gerät spezifiziert, wenn die Datei dem ausführenden Programm zugeordnet wird (normalerweise durch eine OPEN-Anweisung). Der restliche Teil des Programms kann ohne zusätzliche Kenntnisse der bestimmten Datei oder des Geräts, auf welches zugegriffen wird, geschrieben werden. Auf diese Art und Weise ist es möglich, Programme geräteunabhängig zu entwickeln. Darüberhinaus ist die Waterloo-Software in der Lage, besondere Einrichtungen bestimmter Geräte zu benutzen. Dabei muß jedoch bedacht werden, daß das Programm unter Umständen nicht wie gewünscht arbeitet, wenn Geräte angesprochen werden, die nicht mit diesen Sondereinrichtungen ausgestattet sind.

Andere Geräte (wie z.B. Drucker oder Bildschirm) verfügen über eingeschränkte Einrichtungen für besondere Zwecke.

Ist der Mikrocomputer an einen großen Zentralrechner angeschlossen, dann wird dieser mit dem Namen 'host' angesprochen. Dateien können auf den Datenträgern des Zentralsystems gespeichert und von dort auch wieder gelesen werden, wenn zur gleichen Zeit gesichert ist, daß dem Zentralrechner ein Kommunikationsprogramm zur Ausführung zur Verfügung steht, welches den Zugriff auf Dateien dieses Systems steuert. Zur Zeit bestehen Kommunikationsprogramme (genannt 'HOSTCM') für die IBM-Anlagen der Serie /1, der Serien 370/303x, 43xx mit VM/CMS und der DEC-Anlagen PDP-11 mit RSTS/E. Diese Programme werden nicht zusammen mit dem MMF 9000 geliefert, können jedoch von Waterloo angefordert werden.

4.2 DATEIARTEN

Eine Datei besteht aus einer Aneinanderreihung einer Anzahl von Datensätzen, wobei jeder Satz aus einer Anzahl von Zeichen besteht. Die Waterloo Software unterstützt drei unterschiedliche Dateiarten, abhängig von den Sätzen, die der Datei zur Verfügung gestellt werden:

1. Text
   Eine 'Text'-Datei besteht aus unterschiedlich langen Sätzen, die nur mit 'druckbaren' Zeichen bestückt sind. Dies ist der Standard-Dateityp, der vom System automatisch gewählt wird, sofern nicht einer der anderen Typen explizit bestimmt wurde.

2. Variabel
   Eine 'Variabel'-Datei besteht aus unterschiedlich langen Sätzen, deren Inhalt aus beliebigen Zeichen zusammengesetzt werden kann. Diese Dateiart sollte dann spezifiziert werden, wenn Daten in den Sätzen enthalten sind, deren Format nicht auf dem Bildschirm angezeigt werden kann.

3. Fixed
   Eine 'Fixed'-Datei besteht aus Sätzen fester Satzlänge, deren Inhalt sich aus beliebigen Zeichen zusammensetzt. Wird durch das Programm der Versuch unternommen, einen Satz mit einer kleineren Satzlänge zu schreiben, dann erweitert das System den Satz automatisch, indem am Ende Leerzeichen hinzugefügt werden.

Die Dateiart wird als 1. Teil des Dateinamens innerhalb Klammern angegeben. Durch Abtrennen der rechtsbündigen Buchstaben können die Dateiarten in Kurzform eingegeben werden:

(t),(te),(tex),(text).

Alle oben genannten Abkürzungen sind als Spezifikation für eine 'Text'-Datei gültig.

4.3 DATEINAMEN

Das allgemeine Format für einen Dateinamen ist wie folgt:

(Dateiart:Satzlänge)Gerät.Dateidesignator

Die folgenden Standardwerte werden eingesetzt, wenn keine Spezifikation erfolgt:

- der Standardwert für die Dateiart ist 'Text';
- der Standardwert für die Satzlänge ist 80 Zeichen je Satz für Dateien mit festen Satzlängen (fixed); für die anderen Dateiarten wird kein Eintrag benötigt;
- der Standardwert für die Gerätenummer ist '8';
- der Dateidesignator wird standardmäßig nicht eingesetzt.

Das Format des Dateidesignators hängt von dem Datenträger ab, auf welchem sich die Datei befindet. Die entsprechenden Informationen werden zusammen mit den einzelnen Geräten gegeben. Ein Dateidesignator wird nicht benötigt, wenn das Gerät nicht für die Speicherung einer Datei eingesetzt wird (z.B. einem Drucker).

Die Option 'Satzgröße' innerhalb des Dateinamens gibt die maximale Anzahl der Zeichen eines Datensatzes für eine Datei. Bei dem Versuch, einen Satz zu schreiben, der länger als der angegebene Maximalwert ist, wird vom System der überhängende Teil automatisch abgetrennt. Wird die Satzgröße nicht spezifiziert, dann wird im Zusammenhang mit 'Variablen'- oder 'Text'-Dateien nicht auf den Maximalwert überprüft.

Bei den meisten Datenträgern, auf denen sich Dateien befinden, genügt eine einfache Sequenz alphabetischer Zeichen als Dateidesignator:

```
host.myfile
disk.student
(fixed:30)host.data
(text)data
```

Die oben genannten Beispiele zeigen verschiedene Dateinamen. Bei einigen Geräten können Groß- und Kleinbuchstaben spezifiziert werden. Bei anderen Geräten ist nur die Großschreibung erlaubt.

KAPITEL 5

DATEIEN DES

COMMODORE MMF 9000

5.1 EINLEITUNG

Dieses Kapitel beschreibt Dateien von Datenträgern, die über Geräte gelesen oder geschrieben werden, die über einen direkten Anschluß zum MMF 9000 verfügen. Dateien, die in einem Zentralrechner gespeichert werden, werden hier nicht erklärt. Wie bereits im vorhergehenden Kapitel beschrieben, ist das allgemeine Format des Dateinamens wie folgt:

(Dateiart:Größe)Gerät.Dateidesignator

Innerhalb dieses Kapitels wird der 'Dateidesignator' beschrieben und der Einfachheit halber 'Dateiname' genannt.

5.2 Diskettendateien - Allgemein

Die von Commodore für dieses System gelieferte Disketteneinheit besteht aus zwei Laufwerken. Jedes Laufwerk kann eine Diskette aufnehmen. Es ist möglich, auf jede Diskette mehrere Dateien zu speichern, wobei jede Datei mit einem eindeutig individuellen Dateinamen angesprochen werden muß. Der Dateidesignator kann bis zu 16 Zeichen lang sein und kann Leerzeichen und andere Sondezeichen enthalten:

disk.myfile
(text)disk.datafile

Bei den zwei vorhergehenden Beispielen handelt es sich um einfache Dateinamen, deren Dateien sich auf Diskette (disk) befinden.

Es ist möglich, mehrere Disketteneinheiten an ein System anzuschließen. Jede Einheit wird durch eine Nummer oder Adresse angesprochen. Unter normalen Umständen erhält die erste angeschlossene Disketteneinheit die Adresse 8. Die zwei Diskettenlaufwerke einer jeden Disketteneinheit erhalten die Nummern 0 und 1, wodurch das rechte bzw. linke Diskettenlaufwerk angesprochen wird:

disk8/0.myfile
disk8/0.datafile
disk9/1.goophy

Die oben aufgeführten, ersten zwei Zeilen entsprechen den im vorhergehenden Absatz genannten Beispielen. Die letzte Zeile stellt als Beispiel eine Datei mit dem Namen 'goophy' dar, deren Diskette sich in dem linken Laufwerk der zweiten Disketteneinheit befindet.

Das allgemeine Format eines Dateinamens ist daher:

Diskette / Laufwerk . Dateidesignator,Format

wobei

1. 'disk' eine Disketteneinheit anspricht. Die zuerst angeschlossene Disketteneinheit erhält die Adresse 'disk' oder 'disk8'. Wird keine Angabe gemacht, dann nimmt das System den Standardwert 'disk';
2. bei dem Begriff 'Laufwerk' bezieht sich auf die Nummer 0 auf das rechte Laufwerk und die Nummer 1 auf das linke Laufwerk. Wird keine Angabe gemacht, dann setzt das System den Standardwert 0 ein;
3. der Begriff 'Dateidesignator' besteht aus einer Zeichenkette von 1 bis 16 beliebigen Zeichen, einschließlich Leerzeichen und Sonderzeichen;
4. bei dem Teil 'Format' sollte entweder 'seq' (sequentielle Dateiorganisation) oder 'rel' (relative Dateiorganisation) angegeben werden. Wird keine Angabe gemacht, dann setzt das System den Standardwert 'seq' ein.

Der Unterschied zwischen 'rel' und 'seq' besteht in dem Format, in dem Daten auf der Diskette gespeichert werden und in den Eingabe/Ausgabe-Funktionen, die im Zusammenhang mit diesen Formaten eingesetzt werden können. Jede einzelne Dateiart (Text, feste Satzlänge, variable Satzlänge) kann in Verbindung mit beiden Dateiformaten ('seq' oder 'rel') verwendet werden.

Mit dem Format 'seq' werden die Datensätze unmittelbar hintereinander gespeichert. Bei 'Text'-Dateien wird ein jeder Datensatz mit einem Sonderzeichen beendet, welches das Ende eines Satzes anzeigt. Bei 'variabel'-Dateien werden zwei zusätzliche Zeichen dem Anfang eines jeden Datensatzes hinzugefügt, deren Wert die Länge des Datensatzes angibt. Bei 'fixed'-Dateien (Dateien mit festen Satzlängen) werden die Datensätze ohne Hinzufügung zusätzlicher Zeichen gespeichert.

Bei Verwendung des 'rel'-Formates werden die Datensätze entsprechend der Beschreibung des vorhergehenden Absatzes gespeichert. Es wird jedoch in jedem Fall so viel Platz reserviert, wie notwendig ist, um unter Umständen den größtmöglichen Datensatz aufnehmen zu können. Das bedeutet, daß 'rel'-Dateien mehr Platz in Anspruch nehmen, als eine gleichgroße 'seq'-Datei. Wird die Satzgröße nicht spezifiziert, dann setzt das System den Standardwert von 80 Zeichen für die Satzlänge ein.

Der Vorteil des 'rel'-Formats ist der, daß es möglich ist, auf alle Datensätze innerhalb dieser Datei direkt zugreifen zu können. Das bedeutet, daß z.B. der 9. der 2. und der 14. Satz mit lediglich 3 Leseoperationen gelesen werden können. Nach der Speicherung von Datensätzen im 'seq'-Format muß beim Lesen der entsprechenden Datensätze die ursprünglich eingehaltene Reihenfolge der Speicherung berücksichtigt werden.

5.3 DISKETTENDATEIEN - ERWEITERUNGEN

Bei der OPEN-Funktion wird jede Diskettendatei einem Kanal oder einer Sekundäradresse zugeordnet. Obwohl nur bis zu fünf Dateien gleichzeitig für eine Disketteneinheit eröffnet werden können, sind 16 mögliche Kanäle vorhanden, die von 0 bis 15 numeriert sind. Dabei ist der Kanal 0 für Systemoperationen, wie z.B. für die Auflistung der Dateiverzeichnisse reserviert. Der Kanal 1 ist für Operationen, wie z.B. für den STORE-Befehl von BASIC reserviert. Der Kanal 15 wird für Leseoperationen der Fehlerstatusmeldungen von Disketteneinheiten und das Abschicken besonderer Befehle an die Disketteneinheiten benutzt. Die Verwendung des Kanals 15 im Zusammenhang mit diesen besonderen Diskettenbefehlen wird in einem der folgenden Kapiteln beschrieben. Die Kanäle 2-14 stehen für die allgemeine Benutzung in Verbindung mit den Dateien zur Verfügung.

Es folgt das allgemein gültige Format einer Spezifikation eines Diskettendateinamens:

Diskettenadresse - Kanal / Laufwerk . Dateidesignator,Format

wobei die Komponenten wie folgt definiert sind:

Diskettenadresse	= Disketteneinheitadresse (0 - 30),
Kanal	= Dateikanal (0 - 15),
Laufwerk	= Laufwerknummer (0 oder 1),
Dateidesignator	= eine Zeichenkette von 1 - 16 Zeichen,
Format	= Dateiformat ('seq', 'rel').

Werden keine Angaben zu einem bestimmten Teil innerhalb des Gesamtbegriffes gemacht, dann setzt das System automatisch den entsprechenden Sandardwert ein.

Parameter	**Standardwert**
Adresse	8
Kanal	automatisch zugeordnet
Laufwerk	0
Dateidesignator	kein Standardwert
Format	seq

Die folgenden Beispiele geben einige Dateinamen und deren Bedeutung:

Dateiname	**Bedeutung**
charlie	datei 'charlie' auf Laufwerk 0 der Disketteneinheit 8
disk.charlie	wie oben
disk/1.charlie	Datei 'charlie' in Laufwerk 1 der Disketteneinheit 8
disk9.charlie	Datei 'charlie' in Laufwerk 0 der Disketteneinheit 9
disk	Fehler/Befehls-Kanal der Disketteneinheit 8
disk-15	wie oben
(f:80)kunde,rel	relative Datei 'kunde' in Laufwerk 0 der Disketteneinheit 8
(f)kunde, rel	wie obige Datei; Satzgröße nicht für bestehende Datei spezifiziert (Standardwert = 80).

Es ist hierbei zu bemerken, daß der Befehlskanal 15 automatisch genommen wird, wenn kein Dateidesignator spezifiziert wird.

Die Möglichkeit, die Kanalnummer zu spezifizieren, wird denjenigen Benutzern zur Verfügung gestellt, die mit ihrem System besondere Probleme lösen wollen und über detaillierte Kenntnisse der Disketteneinheiten und des IEEE-488-Bus verfügen. Unter normalen Umständen wird empfohlen, diesen Parameter nicht zu spezifizieren.

5.4 DRUCKER

Der von Commodore gelieferte Drucker muß als besondere Datei betrachtet werden, auf die Daten nur ausgegeben werden können. Es ist lediglich der Gerätename

printer

Als Dateiname zu spezifizieren. Die normalen Druckvorgänge können unter Verwendung dieses Dateinamens ausgeführt werden.

5.5 TASTATUR DES BILDSCHIRMTERMINALS

Die Tastatur des Bildschirmterminals kann als Sondergerät mit dem Namen 'keyboard' betrachtet werden. Bei Betätigung der Tasten werden die entsprechenden Zeichen durch die Waterloo Software in den Tastaturpuffer gebracht. Beim Lesen des Gerätes 'keyboard' werden die Daten aus dem Tastaturpuffer übertragen und im Tastaturpuffer selbst gelöscht. Bei leerem Puffer wird ein einzelnes Zeichen mit dem hexadezimalen Wert '00' an das Programm übertragen.

Werden vom Programm an das Gerät 'keyboard' Daten übertragen, dann werden diese den sich im Puffer befindenden Daten hinzugefügt. Dadurch wird es möglich, die Eingabe von Befehlen und/oder Daten an das Programm zu simulieren, als ob diese vom Benutzer eingegeben worden seien.

Der Tastaturpuffer kann 40 Zeichen aufnehmen. Ist der Puffer aufgefüllt, dann werden alle anderen, an das Gerät 'keyboard' übergebenen Zeichen ignoriert.

5.6 BILDSCHIRMTERMINAL

Das Bildschirmterminal kann als Sondergerät mit dem Namen 'terminal' betrachtet werden. Beim Schreiben von Daten werden diese, beginnend bei der derzeitigen Cursor-Position auf den Bildschirm gebracht.

Beim Lesen von diesem Gerät wird der im Bildschirmpuffer vorhandene Inhalt so lange übernommen, bis ein Zeichen registriert wird, welches der RETURN-Taste entspricht. Durch diese Vorgehensweise werden die entsprechenden Zeichen auf dem Bildschirm angezeigt, ausgehend von der Position, an der sich der Cursor zur Zeit befindet. Beim Erreichen eines RETURN-Zeichens werden die Zeichen derjenigen Zeile an das Programm übergeben, die durch den Cursor angezeigt wird. Die sich am Ende eine Zeile eventuell befindenden Leerzeichen werden nicht übertragen.

5.7 SERIELL BETRIEBENE GERÄTE

Der MMF 9000 verfügt über eine Anschlußbuchse, die den Anschluß seriell betriebener ASCII-Geräte, wie z.B. Drucker oder Terminals oder Zentralrechner erlaubt. Dieser Anschluß kann für die Datenübertragungsleitung von und zum Zentralrechner benutzt werden, wie in den folgenden Kapiteln beschrieben, wenn der Rechner am anderen Ende der Datenübertragungsleitung mit der gleichen Schnittstelle ausgerüstet ist. Darüber hinaus ist es möglich, diese Anschlußbuchse für Hilfsgeräte, wie z.B. ASCII-Drucker oder nicht-intelligente Terminals zu verwenden. Der Begriff 'nicht-intelligente Terminals' betrifft Geräte, die nicht zur Steuerung des Ablaufs eingesetzt werden, sondern nur als zusätzliche Eingabe/Ausgabe-Geräte zur Verfügung stehen.

Es ist hierbei zu bemerken, daß der von Commodore gelieferte Drucker nicht an diese Anschlußbuchse angeschlossen werden kann. Ähnlich der von Commodore gelieferten Disketteneinheit kommuniziert der Drucker mit dem System über ein IEEE-488-Protokoll und wird aus diesem Grund mit dem IEEE-488-Datenbus verbunden.

Wird der Anschluß in Verbindung mit einem Hilfsgerät benutzt, dann ist der Gerätename wie folgt:

    serial

So z.B. kann ein ASCII-Drucker mit dieser Anschlußbuchse verbunden werden, wobei der Dateiname 'serial' Verwendung findet.

5.8 IEEE-488-GERÄTE

Der MMF 9000 ist mit einem Datenbusanschluß IEEE-488 ausgerüstet, an den die von Commodore gelieferten Disketteneinheiten und Drucker angeschlossen werden. Die Dateinamen zur Spezifikation dieser Geräte wurde in vorausgehenden Abschnitten beschrieben. Ferner stehen andere Gerätetypen zur Verfügung, wie z.B. Plotter, Labormeßgeräte, usw., die dem System unter Verwendung des IEEE-488-Bus angeschlossen werden können. Zur Adressierung dieser Geräte, die mit dem System verbunden sind, ist der Dateiname im folgenden Format zu spezifizieren:

IEEE Adresse - Kanal

wobei sich Adresse auf die Primäradresse des Geräts (0 - 30) und Kanal auf die Sekundäradresse (0 - 30) bezieht.

Die Standardwerte für sowohl 'Adresse' als auch 'Kanal' sind = 0. Der folgende Beispieldateiname bezieht sich auf ein IEEE-488-Gerät mit der Primäradresse 10 und der Sekundäradresse (Kanal) 1:

ieee10-1

Die Software des Waterloo Mikrosystems übernimmt die Verarbeitung des Datenbus-Protokolls IEEE-488 für diese Art der Datenkommunikation. Der Programmierer muß sich jedoch um besondere Anforderungen im Zusammenhang mit den benutzten Geräten kümmern. Es ist hierbei zu bemerken, daß die von Commodore gelieferte Disketteneinheit und der Drucker nicht mit diesem Typ des Dateinamens angesprochen werden sollten, wenn es gewünscht ist, daß die Software automatisch die besonderen Merkmale dieser Geräte berücksichtigt.

5.9 UNTERSTÜTZUNG DER COMMODORE DISKETTENBEFEHLE

Die mit diesem System von Commodore gelieferten Disketteneinheiten sind mit Mikroprozessoren ausgerüstet, die eine Anzahl von Befehlen zur Verwaltung der Disketten unterstützen. Die entsprechenden Befehle werden in dem jeweiligen, zusammen mit der Disketteneinheit gelieferten Handbuch beschrieben. Da diese Commodore-Handbücher für die Benutzung der Disketteneinheiten mit Commodore BASIC erstellt wurden, muß der Leser die Beispiele den jeweiligen Gegebenheiten anpassen, wie unten dargestellt. Diese Befehle können von Programmen benutzt werden, indem ein Befehl zur Eröffnung des Kanals 15 der entsprechenden Disketteneinheit als Datei abgesetzt wird und darauf folgend Befehlszeichenketten auf die Sonderdatei geschrieben werden.

Obwohl die Anweisungen des Waterloo MicroBASIC benutzt wurden, um den Einsatz der Diskettenbefehle zu demonstrieren, kann die gleiche Vorgehensweise zum Einsatz anderer Waterloo Micro-Sprachen verwendet werden.

Das folgende Beispiel demonstriert die Benutzung des VALIDATE-Diskettenbefehls, womit eine 'Block Availability Map' erstellt werden soll, um das Verzeichnis des Diskettenlaufwerks 0 der Einheit 9 zu überprüfen.

```
open #2,'disk9',output
print #2,'VALIDATE 0'
```

In diesem Beispiel spricht '#2' eine Dateinummer an, die der Datei durch OPEN-Anweisung zugeordnet wurden. Die Dateinummer wird in der PRINT-Anweisung zur Adressierung der Datei verwendet.

Ähnliche Methoden können angebracht werden, um andere Befehle auf die Diskette abzusetzen. Dabei ist lediglich der Begriff 'VALIDATE 0' mit dem entsprechenden Befehl zu ersetzen.

ANMERKUNG:

Bestimmte Commodore-Diskettenbefehle (z.B. INITIALIZE, NEW, DUPLICATE oder VALIDATE) haben zur Folge, daß die Disketteneinheit alle über sie verfügbaren Dateien abschließt, obwohl das Computersystem diese immer noch als 'eröffnet' betrachtet. Derartige Operationen sollten nicht ausgeführt werden, wenn sich noch einige Dateien dieser Disketteneinheit im 'OPEN'-Zustand befinden. Im Gegensatz zum Commodore BASIC bedeutet der Abschluß des Befehlskanals (CLOSE) nicht, daß die Disketteneinheit alle darüber verfügbaren Dateien abschließt.

5.10 COMMODORE DRUCKER-FORMATIERUNGSUNTERSTÜTZUNG

Der mit diesem System eingesetzte Commodore Drucker verfügt über einen Mikroprozessor, welcher die Ausgabeformatierung unterstützt und die Definition neuer Zeichen erlaubt. Ähnlich einer Disketteneinheit verfügt der Drucker über Kanäle und Sekundäradressen, über die Daten empfangen werden können. Der Dateiname 'printer' wird benutzt, wobei Daten über den Kanal 0 übertragen werden. In diesem Fall werden die Daten genau so gedruckt, wie sie auch zum Drucker abgesendet wurden. Andere Kanäle stellen Sonderfunktionen zur Verfügung, wie diese in dem zusammen mit dem Drucker gelieferten Handbuch von Commodore beschrieben sind. Da das Commodore-Handbuch unter Berücksichtigung des Commodore BASIC geschrieben wurde, muß der Leser die Befehle entsprechend der unten aufgeführten Beispiele ändern. Die Kanäle können für die Ausgabe eröffnet werden, indem der Dateiname im folgenden Format spezifiziert wird:

printer Adresse - Kanal

wobei Adresse die Druckeradresse ist (0 - 30) und Kanal eine Nummer zwischen 0 und 6 (Modell 2022) oder 10 (Modell 4022) erhält.

Unter normalen Umständen wird an den MMF 9000 nur ein Drucker über die Adresse 4 angeschlossen. Daher wird vom System die Adresse 4 als Standardwert automatisch angenommen, wenn der Dateiname keine entsprechenden Angaben enthält. Der Commodore Drucker wird über den IEEE-488-Datenbus mit dem System verbunden. Ab und zu werden auch andere Begriffe für 'Adresse' und 'Kanal' benutzt; nämlich 'Primäradresse' und 'Sekundäradresse'.

Das folgende Beispiel illustriert die Definition einer Formatierungs-Zeichenkette im Zusammenhang mit Waterloo MicroBASIC unter Verwendung des Druckerkanals 2:

```
open #5,'printer-2',output
print #5,'$99.99'
close #5
```

In diesem Beispiel ist '#5' die Dateinummer, die der Datei bei der Eröffnung (OPEN) zugeordnet wird.

Commodore verfügt über ein eigenständiges Kodierschema für Groß- und Kleinbuchstaben, eckige Klammern, Rückschrägstrich, Obenpfeil und Unterstreichung. Dieses Kodierschema steht nicht im Einklang mit dem standardisierten 7-Bit ASCII-Code, welcher von der Waterloo Microsystem-Software benutzt wird. Die Waterloo-Software übersetzt diese Zeichen automatisch, so daß der Druck des Zeichens entsprechend der Eingabe über die Tastatur und der Ausgabe auf Bildschirm erfolgt.

ANMERKUNG:

Ein Nebeneffekt dieser Übersetzung besteht darin, daß die Sonderformatzeichen A, Z und S, die für die Definitionen der Feldformatierung als Ausgabe über Kanal 2 benutzt werden, mit deren Kleinschreibungsäquivalent a, z und s ersetzt werden müssen.

KAPITEL 6

DATEIVERWALTUNG - ZENTRALRECHNER

6.1 EINLEITUNG

Wie bereits erwähnt ist es möglich, den Commodore MMF 9000 an einen Zentralrechner anzuschließen, damit auf die dort verwalteten Dateien zugegriffen werden kann. Dazu ist es notwendig, daß im Zentralrechner ein Datenübertragungsprogramm (HOSTCM) läuft, welches den Zugriff auf diese Dateien steuert. Dieses Programm wird abhängig von den verschiedenen Zentralrechnern unterschiedliche Merkmale aufweisen und kann daher nicht als Teil der Commodore-Software geliefert werden.

6.2 KOMMUNIKATION MIT DEM ZENTRALRECHNER

Um zu gewährleisten, daß der MMF 9000 unter Verwendung eines Datenübertragungsprotokolls entsprechend den Bedingungen des Zentralrechners Daten abrufen kann, verfügt der Benutzer über das Hauptmenü der Waterloo-Software über ein Host-SETUP-Programm. Die Benutzung dieses Programms wird in einem der folgenden Kapiteln beschrieben.

6.3 DATEIEN DES ZENTRALRECHNERS

Die Waterloo-Programmiersprachen und der MicroEditor ermöglichen es, auf die Dateien des Zentralrechners so zuzugreifen, als ob diese Dateien sich auf Datenträgern befänden, die über Peripheriegeräte gelesen werden, welche dem MMF 9000 unmittelbar angeschlossen sind. Die Programmiersprachen erfordern, daß ein Dateiname spezifiziert wird (normalerweise im Zusammenhang mit der OPEN-Anweisung) bevor auf diese Datei zugegriffen wird. Das Format des Dateinamens ist wie folgt:

```
(Dateiart:Größe)host.Dateidesignator
```

wobei die Optionen 'Dateiart' und 'Größe' im Kapitel 4 beschrieben sind. Der Parameter 'Zentralrechner' zeigt an, daß die Datei bei dem Zentralrechner zu finden ist. Der Dateidesignator-Teil repräsentiert den Dateinamen im Format des Zentralrechners. Es würde den Rahmen dieses Handbuches sprengen, das Format für jeden möglichen Zentralrechner hier anzugeben. Dieses sollte unter Beachtung der entsprechenden Dokumentation für den Zentralrechner ausgearbeitet werden.

Unter normalen Umständen nehmen die meisten Zentralrechner einen Dateinamen von mindestens 6 Zeichen (Großbuchstaben) an:

```
host.DATA
(f:55)host.SPECS
(t)host.MYPROG
```

Bei den hier genannten Beispielen handelt es sich um typische Zentralrechner-Dateinamen.

KAPITEL 7

WATERLOO-MICROEDITOR

7.1 EINLEITUNG

Der Waterloo-MicroEditor ist ein Texteditor, der sich für die Erstellung und Bearbeitung von Quellenprogrammen und Daten eignet. Der Editor basiert auf Bildschirmverarbeitung und benutzt die zur Verfügung stehenden Programmfunktionstasten (PF). Dieses Kapitel besteht aus zwei Hauptabschnitten:

1. mit dem Lehrmaterial zur MicroEditor-Handhabung (siehe Abschnitt 7.2) wird versucht, eine kurze Einleitung zur Handhabung des MicroEditor zur Verfügung zu stellen;
2. die Abschnitte 7.3 und 7.4 geben genaue Angaben über die einzelnen Einrichtungen des MicroEditors.

7.2 LEHRMATERIAL MICROEDITOR-HANDHABUNG

Es ist von besonderer Bedeutung, daß der Leser beim Durcharbeiten dieses Abschnitts die Beispiele an einem MMF 9000 selbst nachvollzieht, um sicherzustellen, daß er sie vollständig verstanden hat.

7.2.1 Einleitung

Der Waterloo MicroEditor kann direkt über das Hauptmenü gestartet werden. Zusätzlich ist der Editor in die Waterloo-Software MicroFORTRAN und MicroPASCAL integriert, was bedeutet, daß er durch Selektion dieser Programmiersprachen über das Hauptmenü automatisch mit aufgerufen wird. Ferner ist es möglich, unter Verwendung des EDIT-Befehls des Waterloo MicroBASIC den MicroEditor aufzurufen.

Unmittelbar nach dem Laden des MicroEditors ist der Arbeitsbereich leer, was bedeutet, daß sich keine Zeilen zwischen <Dateianfang> und <Dateiende> befinden.

Nach dem Start des MicroEditors erscheinen zwei Positionsanzeiger (Cursor) auf dem Bildschirm. Bei dem oberen Block handelt es sich um den 'Text-Cursor' und bei dem unteren Block um den 'Befehls-Cursor'. Der Text-Cursor steht auf der derzeitig aktiven Zeile.

Viele der Editorfunktionen werden durch Betätigung einer der 11 PF-Tasten (Programmfunktionstasten) - PF0 bis PF9 und PF. - aufgerufen. Diese Tasten werden aktiviert, indem die Umschalttaste, zusammen mit der entsprechenden numerischen Taste des Nummernblockes (0-9) gedrückt wird. So z.B. wird PF8 aktiviert, indem die Umschalttaste gehalten und die Zifferntaste 8 des Nummernblocks gedrückt wird.

Unter Verwendung der Links-Rechts-Cursor-Steuertasten wird der Cursor nach links oder nach rechts bewegt, um die Korrektur von Daten zu ermöglichen. Auch kann die TAB-Taste dazu verwendet werden, den Cursor nach rechts an das Zeilenende springen zu lassen. Bei Betätigung der INST-Taste werden für die Eingabe zusätzlicher Zeichen auf der gleichen Zeile Leerstellen geschaffen. Wird die DEL-Taste gedrückt, dann werden Zeichen von der Zeile entfernt. Die Überschreibungstaste (REPEAT) ersetzt di Zeichen durch Leerzeichen. Die Taste 'Löschen bis zum Ende der Zeile' (ESC) bewirkt genau das, was durch die Bezeichnung ausgesagt wird. Die Betätigung der RUN-Taste führt zur Ausführung des Inhalts des Arbeitsbereichs, wenn der Editor in Verbindung mit einem Programmsprachen-Prozessor eingesetzt wird. Unter normalen Umständen schließt die Betätigung der STOP-Taste die derzeitig laufende Editorfunktion ab.

Alle Tasten sind mit der Dauerfunktion ausgerüstet, wodurch die entsprechende Funktion wiederholt ausgeführt wird, wenn die Taste gedrückt gehalten wird.

7.2.2 Dateneingabe

Die Aktivierung von PF0 führt zur Erstellung einer neuen Zeile, in die der gewünschte Text eingegeben werden kann. Der Benutzer sollte das Anlegen neuer Textzeilen üben.

Nach Aktivierung der PF0-Taste verschwindet der Befehls-Cursor vom Bildschirm. Alle unmittelbar darauf eingegebenen Zeichen erscheinen in der neu erstellten Zeile. Wenn auf dem Bildschirm kein Befehls-Cursor zu finden ist, befindet sich der MicroEditor im 'Bildschirmmodus'.

7.2.3 Änderungen

Um einen Text zu ändern, ist es lediglich notwendig, die Taste CRSR (hoch/runter) zu drücken und somit den Cursor auf die entsprechende Zeile zu bringen. Danach kann jeder beliebige Teil dieser Zeile mit anderen Zeichen überschrieben werden. (Die Tasten PF7 und PF4 führen zu den gleichen Funktionen, wie die Taste CRSR (hoch) und CRSR (runter).) In Verbindung mit Änderungen zum Text sind auch die Tasten CRSR (Linkspfeil), CRSR (Rechtspfeil), INST und DEL besonders nützlich.

Befindet sich das System im Bildschirmmodus und wird die STOP-Taste gedrückt, dann wird die derzeitig angesprochene Zeile in ihren ursprünglichen Zustand versetzt. Mit dem 'ursprünglichen Zustand' ist der Zustand gemeint, der vorhanden war, bevor der Cursor zum letzten Mal auf diese Zeile geführt wurde. Diese Taste ist dann besonders nützlich, wenn Änderungen erfolgt sind, die sich im Nachhinein als nicht notwendig erwiesen.

Es ist hierbei zu bemerken, daß eine neue Zeile nach einer beliebigen, bereits bestehenden Zeile hinzugefügt werden kann, indem der Cursor auf die entsprechende Zeile geführt, PF0 aktiviert und die neue Zeile eingegeben wird.

7.2.4 Zeilenlöschung

Für die Löschung einer Zeile ist der Cursor unter Verwendung der Taste CRSR (hoch/runter) auf die zu löschende Zeile zu führen und danach PF2 zu aktivieren.

7.2.5 Befehlsmodus

Viele der Funktionen des MicroEditors werden aufgerufen, indem im Unterschied Verwendung einer PF-Taste ein Befehl eingegeben wird. Zur Eingabe dieses Befehls ist es notwendig, den Editor in den 'Befehlsmodus' zu versetzen. Befindet sich nur ein Cursor auf dem Bildschirm, dann ist der 'Bildschirmmodus' eingeschaltet. Zum Einschalten des 'Befehlsmodus' ist die Taste PF5 zu aktivieren. Ein zweiter Cursor wird darauf in der Befehlszeile erscheinen. Alle durch die PF-Tasten ausgelösten Funktionen sind auch im Befehlsmodus verfügbar.

Der Editor kann in den Bildschirmmodus versetzt werden, wenn die PF8-Taste aktiviert wird. Darüber hinaus wird der Bildschirmmodus eingeschaltet, wenn durch die Betätigung der PF0-Taste eine neue Zeile erzeugt wird.

7.2.6 Positionierung auf Verarbeitungszeile

Der MicroEditor kann auf eine neue Zeile gesetzt werden, indem lediglich die Nummer der gewünschten Zeile eingegeben wird. So z.B. erreicht man durch die Eingabe von '6' die 6. Zeile des zur Zeit bearbeiteten Arbeitsbereiches. Die Eingabe von '0' führt zu <Dateianfang>. Für die Angabe der letzten Zeile einer Datei ist das Dollarzeichen ($) einzugeben. Es können auch einfache arithmetische Formeln benutzt werden. So z.B. erreicht die Eingabe $-6 die siebtletzte Zeile einer Datei. Darüber hinaus ist es möglich, den Cursor auf eine bestimmte Zeile relativ zur derzeitigen Position zu bringen, indem entweder das Pluszeichen (+) oder Minuszeichen (-), gefolgt von einer Nummer eingegeben wird. So z.B. wird der Cursor auf die 6. Zeile nach der z. Zt. aktiven Zeile geführt, wenn +6 eingegeben wird. Ferner ist es möglich, einige PF-Tasten zur Positionierung des Cursors auf bestimmte Zeilen zu verwenden:

PF6 = eine Zeile zurück (Befehl = -1),
PF3 = eine Zeile vorwärts (Befehl = +1),
PF9 = eine Bildschirmmaske zurück (Befehl = -20),
PF. = eine Bildschirmmaske vorwärts (Befehl = +20).

7.2.7 Auffinden von Zeilen durch Suchbegriff

Der MicroEditor ist in der Lage, den Arbeitsbereich nach einer Zeichenkette zu durchsuchen. So z.B. führt die Eingabe von

/Text

dazu, daß die erste Zeile, in welcher die Zeichenkette 'Text' enthalten ist, als derzeitig zu bearbeitende Zeile angezeigt wird.

Die Zeile, die unmittelbar darauf die Zeichenkette 'Text' enthält, wird dadurch erreicht, daß die folgende Eingabe erfolgt:

+/Text

Auf ähnliche Weise kann die unmittelbar davor liegende Zeile, die diese Zeichenkette enthält, erreicht werden:

-/Text

7.2.8 Änderungen zum gesamten Arbeitsbereich

Es ist oft wünschenswert, daß eine bestimmte Zeichenkette innerhalb des Arbeitsbereichs gegen eine andere Zeichenkette ausgetauscht wird. So z.B. ist es möglich, daß das Wort 'Zeile' in das Wort 'Zeichen' umzuändern ist. Das wird dadurch erreicht, daß folgender Befehl eingegeben wird:

c/Zeile/Zeichen

7.2.9 Löschen des Arbeitsbereiches

Der Arbeitsbereich kann mit folgendem Befehl gelöscht werden:

*d

Es sollte hierbei bemerkt werden, daß durch die Eingabe des einfachen Löschbefehls

```
d
```

nur die derzeitig bearbeitete Zeile gelöscht wird.

7.2.10 Sichern des Arbeitsbereiches

Ist der Editvorgang für den Inhalt eines Arbeitsbereiches abgeschlossen, dann muß dieser Inhalt auf einer Diskettendatei gesichert werden. Diese Sicherung erfolgt durch die Eingabe des 'PUT'-Befehls. So z.B. wird der Inhalt des Arbeitsbereichs unter Verwendung des Befehls

```
p mytext
```

auf der Diskette unter dem Namen 'mytext' gespeichert. Nach Abschluß der Operation erscheint eine Nachricht, welche die Anzahl der gesicherten Zeilen angibt.

Die Datei kann wieder eingelesen werden, indem der Arbeitsbereich zuerst mit dem Befehl '*d' gelöscht wird, wonach der 'GET'-Befehl einzusetzen ist.

```
*d
g mytext
```

7.3 BILDSCHIRMANZEIGE UND PF-TASTEN

Die Bildschirmanzeige stellt sogenannte 'Fenster' zur Verfügung, durch die es möglich ist, sich eine Anzahl von Zeilen aus Dateien zu betrachten. Unter Verwendung der Programmfunktionstasten ist es möglich, den Text zu ändern, neuen Text hinzuzufügen oder Textteile direkt am Bildschirm zu löschen.

Der Bildschirm ist in drei Bereiche aufgeteilt:

1. ein 'Fensterausschnitt' der Datei,
2. eine Zeile für die Befehlseingabe,
3. eine Zeile für die Ausgabe von Nachrichten.

Der Bildschirmaufbau ist unten dargestellt.

```
+------------------------------------------------------------------+
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:                 'FENSTERAUSSCHNITT' DER DATEI                    :
:                        (Textbereich)                             :
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:                                                                  :
:------------------------------------------------------------------:
:                    Befehlseingabe-Bereich                        :
:------------------------------------------------------------------:
:                  Nachrichtenausgabe-Bereich                      :
+------------------------------------------------------------------+
```

Eine der im Textbereich angezeigten Zeilen gilt durch die Stellung des Cursors als derzeitig bearbeitete Zeile. Üblicherweise beziehen sich die Editor-Befehle auf die derzeitig bearbeitete Zeile, wie im Abschnitt 7.4 beschrieben.

Die am häufigsten benutzten Editor-Befehle sind über die Programmfunktionstasten (PF) anwählbar. Eine PF-Taste wird aktiviert, indem zuerst die Umschalttaste gedrückt und gehalten wird und danach die entsprechende PF-Taste. Der Nummernblock ist wie folgt ausgelegt:

7 Verschiebung der derzeitig bearbeiten Zeile um eine Zeile nach oben	8 Umschaltung in den Bildschirmmodus	9 Verschiebung des Fensterausschnitts um einen Bildschirm nach oben
4 Verschiebung der derzeit bearbeiteten Zeile um eine Zeile nach unten	5 Umschaltung in den Befehlsmodus	6 Verschiebung des Fensterausschnitts um eine Zeile in der Datei nach oben
1	2 Löschung der derzeitig bearbeiteten Zeile	3 Verschiebung des Fensterausschnitts um eine Zeile in der Datei nach unten
Ø Einfügen einer Leerzeile hinter der derzeitig bearbeiteten Zeile		. Verschiebung des Fensterausschnitts um einen Bildschirm nach unten

CRSR (hoch/runter) bietet die gleichen Funktionen wie PF7 und PF4.

Der Editor verfügt über zwei Betriebsmodi. Im 'Befehlsmodus' befindet sich ein Cursor im Befehlsbereich, in dem die Befehle eingegeben werden, während die derzeitig bearbeitete Zeile im Textbereich durch den Text-Cursor gekennzeichnet wird. Im 'Bildschirmmodus' ist nur der Text-Cursor vorhanden, den der Benutzer an eine beliebige Stelle innerhalb des Textbereichs führen kann, um dort den Inhalt der Datei zu bearbeiten (Ändern, Einfügen, Löschen). Die PF-Tasten können sowohl im Bildschirmmodus als auch im Befehlsmodus eingesetzt werden. Befehle können nur dann eingegeben werden, wenn sich das System im Befehlsmodus befindet. Durch die Betätigung der Taste PF8 wird das System in den Bildschirmmodus und durch Drücken von PF5 in den Befehlsmodus umgeschaltet.

7.4 BEFEHLS-SYNTAX UND -SEMANTIK

7.4.1 Spezifikation einzelner Zeilen

Eine Zeile einer Datei wird durch ihre "<Zeilen-ID>" angesprochen, wobei die folgenden Angaben möglich sind:

<Zeilennummer>	die durch <Zeilennummer> angesprochene Zeile;
<Zeilennummer>+n	die n-te Zeile nach der angegebenen Zeile;
<Zeilennummer>-n	die n-te Zeile vor der angegebenen Zeile.

Eine <Zeilenangabe> kann aus folgenden Informationen bestehen:

$	letzte Zeile der Datei,
.	derzeitige Zeile,
n	n-te Zeile einer Datei,
+	die unmittelbar nach der derzeitig bearbeiteten Zeile folgende Zeile,
+n	die n-te Zeile nach der derzeitig bearbeiteten Zeile,
-	die unmittelbar vor der derzeitig bearbeiteten Zeile bestehende Zeile,
-n	die n-te Zeile vor der derzeitig bearbeiteten Zeile,
/Text/	die erste Zeile der Datei, die den zwischen den Schrägstrichen angegebenen Suchbegriff enthält,
+/Text/	die erste Zeile, die derjenigen folgt, welche den angegebenen Suchbegriff enthält,
-/Text/	die unmittelbar vor der Zeile liegenden Zeile, welche den angegebenen Suchbegriff enthält.

Die <Zeilenangabe> kann den Wert 0 enthalten, wodurch die derzeitig bearbeitete Zeile angezeigt wird.

7.4.2 Suchbegriffe

Ein Suchbegriff besteht lediglich aus einer Zeichenkette, die vom Editor mit dem Arbeitsbereich oder der derzeitig bearbeiteten Zeile verglichen wird. Üblicherweise werden die Suchbegriffe in Schrägstriche (/) eingeklammert, obwohl der letzte Schrägstrich ausgelassen werden kann, wenn dies auch gleichzeitig das letzte Zeichen der Befehlszeile ist.

Der Sondersuchbegriff '//' stellt den letzten Nicht-Null-Suchbegriff eines Befehls dar.

Einige Zeichensequenzen (als Meta-Zeichen bekannt) haben eine besondere Bedeutung. Diese sind wie folgt:

%/	stellt ein '/' dar,
%↑	stellt den Anfang einer Zeile dar,
%$	stellt das Ende einer Zeile dar,
%.	stellt ein beliebiges Zeichen dar,
%*	stellt eine Null oder eine Wiederholung eines Zeichens dar, welches '%*' unmittelbar folgt,
%%	stellt ein '%' dar.

Ferner besteht ein zusätzliches Meta-Zeichen, welches sich auf die Textbeschreibung eines Änderungsbefehls (siehe Abschnitt 2.6.3) bezieht. Dieses Meta-Zeichen '% ' stellt den Textteil dar, der durch einen Suchbegriff eines Änderungsbefehls gefunden wurde (wenn vorhanden).

Beispiele:

/%↑abc	sucht nach 'abc' am Anfang einer Zeile,
/abc%$	sucht nach 'abc' am Ende einer Zeile,
/a%*bc	stellt eine Null dar oder mehrere 'a', gefolgt von 'bc' ('bc', 'abc', 'aabc', 'aaabc', ...),
/a%.c	stellt 'a' dar, gefolgt von einem beliebigen Zeichen, gefolgt von einem 'c' ('aac', 'abc', 'acc', 'adc', ...),
/ab%.%*d	stellt ein 'ab' dar, gefolgt von einer beliebigen Zeichenkette beliebiger Länge (einschließlich einer leeren Zeichenkette), gefolgt von einem 'd'.

7.4.3 Spezifikation von Zeilenbandbreiten

Viele der Editor-Befehle können sich nicht nur auf eine einzelne Zeile sondern auf eine Anzahl von Zeilen beziehen. Die Befehlsbeschreibung des nächsten Abschnittes benutzen die Begriffe '<Zeilenbandbreite>', wobei dieser Begriff die folgende Bedeutung hat:

<Zeilen-ID>	die durch <Zeilen-ID> angezeigte Zeile;
<Zeilen-ID>,<Zeilen-ID>	alle Zeilen, anfangend mit der ersten <Zeilen-ID> bis zur zweiten <Zeilen-ID>; wenn einer dieser Begriffe ausgelassen wird, dann wird '.' (derzeitig bearbeitete Zeile) angenommen;
*	alle Zeilen in der Datei (gleich der Bandbreite 1,$);
*/Text/	alle Zeilen, die den Suchbegriff 'Text' enthalten;
<Zeilen-ID>↑	die 20 Zeilen, die der <Zeilen-ID> vorausgehen (gleich der Bandbreite '.-20,.'; zeigt den vorhergehenden Bildschirminhalt an, wenn alleinstehend benutzt);
<Zeilen-ID>	zeigt die nach <Zeilen-ID> unmittelbar folgenden 20 Zeilen (gleich der Bandbreite '.,.+20'; zeigt den nächsten Bildschirminhalt an, wenn alleinstehend benutzt).

Beispiele:

5,10	Die Zeilen 5 - 10 einschließlich
,+5	Die derzeitig bearbeitet Zeile und die unmittelbar darauf folgenden 5 Zeilen (könnte auch mit .,.+5 oder .,+5 oder ,.+5 erreicht werden)
-5,+5	Die 5 Zeilen, die der derzeitig bearbeiteten Zeile vorausgehen, die derzeitig bearbeitete Zeile und die 5 Zeilen, die unmittelbar darauf folgen (11 Zeilen insgesamt)
-/abc/,+/def/	Alle Zeilen, ab der ersten Zeile, die den Suchbegriff 'abc' enthält, wenn rückwärts von der derzeitigen Zeile gesucht wird; bis zur ersten Zeile, die den Suchbegriff 'def' enthält, wenn von der derzeitig bearbeiteten Zeile vorwärts gesucht wird;
*/%↑Test/	alle mit dem Suchbegriff 'Test' beginnenden Zeilen.

7.4.4 Befehle

In den folgenden Beschreibungen sind die Angaben von <Zeilen-ID> und <Zeilenbandbreite> wahlfrei. Werden diese nicht spezifiziert, dann wird '.' eingesetzt. Im Syntaxmodell eines jeden Befehls stellen die Kleinbuchstaben die kleinste Abkürzung des jeweiligen Befehls dar. So z.B. lautet das Syntaxmodell für den Verzeichnisbefehl 'diRECTORY', woraus folgt, daß 'di' eine gültige Abkürzung ist.

7.4.4.1 Nummernzeichen

Syntax: #

Beschreibung: Der Nummernzeichenbefehl ("#") führt zur Anzeige der Zeilennummer der derzeitig bearbeiteten Zeile.

7.4.4.2 Fragezeichen

Syntax: ?

Beschreibung: Der Fragezeichenbefehl ("?") ruft den zuletzt eingegebenen Befehl wieder auf, so daß dieser geändert und wieder erneut abgesetzt werden kann.

7.4.4.3 Bye

Syntax: bye

Beschreibung: Der bye-Befehl führt zur Beendigung der derzeitigen Sitzung.

7.4.4.4 Change (Änderung)

Syntax:

```
<Zeilenbandbreite>cHANGE/Text1/Text2/
<Zeilenbandbreite>cHANGEn/Text1/Text2/
<Zeilenbandbreite>cHANGE*/Text1/Text2/
cHANGE
```

Beschreibung: Beim Arbeiten mit dem Editor stehen zwei grundsätzlich unterschiedliche Arten von Änderungsbefehlen zur Verfügung. Der erste Befehlstyp, der in den ersten drei oben aufgeführten Beispielen angezeigt ist, führt zum Austausch des Suchbegriffs 'Text1' mit dem Begriff 'Text2' in allen Zeilen, die durch <Bandbreite> spezifiziert sind. Die erste Zeile bezieht sich lediglich auf die erste Textzeile, in der 'Text1' gefunden wurde, während sich der zweite Befehl auf die Zeile 'n' und der dritte Befehl auf alle Textzeilen bezieht. Die zweite Änderungs-Befehlsart, die in de vierten Zeile angezeigt ist, ermöglicht die Änderung des Bildschirminhalts der derzeitig bearbeiteten Zeile.

Beispiele:

1. c/a/b — Ändert den ersten gefundenen Suchbegriff 'a' in 'b';
2. c*/y/z — Ändert alle Suchbegriffe 'y' in 'z';
3. c — Zeigt die derzeitige Zeile an, worauf Änderungen durchgeführt werden können.

7.4.4.5 Copy (Kopieren)

Syntax:

copY <Eingabe-Dateiname> <Ausgabe-Dateiname>
copY <Eingabe-Dateiname> to <Ausgabe-Dateiname>

Beschreibung: Dieser Befehl kann benutzt werden, um eine Datei (spezifiziert durch <Eingabe-Dateiname>) auf eine andere Datei zu kopieren (spezifiziert durch <Ausgabe-Dateiname>.

Beispiele:

copy disk.test.data disk/1.test.backup
copy source.file, prg to disk9.source.file,prg
copy "data file 1" to "data file 2"

7.4.4.6 Date (Datum)

Syntax:

datE <Datum> <Zeit>
datE

Beschreibung: Mit diesem Befehl ist es möglich, Datum und Zeit zu setzen oder anzuzeigen. Bei Angabe von Datum und Zeit werden diese Informationen in dem System eingegeben. Ohne

Angaben dieser Werte werden diese lediglich aus dem System gelesen und auf dem Bildschirm angezeigt. Das <Datum> kann aus bis zu 11 Zeichen bestehen. Sind im Datum Leerzeichen enthalten, so muß es in Anführungszeichen (") eingeschlossen sein. <Zeit> muß im Format hh:mm:ss:ee angegeben werden, wobei hh = Stunden (hours), mm = Minuten, ss = Sekunden und ee = Einheiten (1/60 einer Sekunde) darstellen.

Beispiele: date 30/06/81 10:45
date "30. Juni 81"

7.4.4.7 Delete (Löschen)

Syntax: <Zeilenbandbreite>dELETE

Beschreibung: Mit dem Löschbefehl können alle Zeilen der spezifizierten Zeilenbandbreite gelöscht werden.

Beispiele:

1. *d	Löschung aller Zeichen
2. 1,10d	Löschung der Zeilen 1-10 einschließlich

7.4.4.8 Directory (Verzeichnis)

Syntax: diRECTORY <Gerätename>

Beschreibung: Der Verzeichnisbefehl listet die Namen alle Dateien auf, die in dem spezifizierten Verzeichnis vorhanden sind.

Beispiele:

di disk	listet das Verzeichnis der Default-Diskette
di disk9/1	listet das Verzeichnis der Disketteneinheit 9, Laufwerk 1

7.4.4.9 Get (Lesen)

Syntax: <Zeilen-ID>gET<Dateiname>
<Zeilen-ID>gET

Beschreibung: Durch den GET-Befehl wird der gesamte Inhalt einer spezifizierten Datei hinter der Zeile eingefügt, die durch <Zeilen-ID> angegeben ist. Wenn kein Dateiname spezifiziert wird, dann nimmt das System den derzeitig benutzten Dateinamen automatisch an. Bei Angabe eines Dateinamens wird dieser als derzeitig benutzter Dateiname für die darauf folgenden Aktivitäten übernommen.

Beispiele: $g demo Fügt die Datei mit dem Namen 'demo' nach der letzten Zeile der Datei hinzu.

7.4.4.10 hELP (Hilfe)

Syntax: hELP

Beschreibung: Dieser Befehl führt zur zusammengefaßten Auflistung der Editor-Befehle auf dem Bildschirm.

7.4.4.11 Input (Eingabe)

Syntax: <Zeilen-ID>iNPUT

Beschreibung: Der INPUT-Befehl setzt das System auf die derzeitig bearbeitete Zeile <Zeilen-ID> und versetzt den Editor in den Eingabemodus. Der nachfolgend eingegebene Text wird der Datei so lange hinzugefügt, bis eine Zeile eingegeben wird, die lediglich einen Punkt (".") enthält.

Beispiele:

1. i Umschalten in den Eingabemodus in der aktuellen Zeile
2. 3i Umschalten in den Eingabmodus nach der 3. Zeile

7.4.4.12 List (Auflistung)

Syntax: <Zeilenbandbreite>

Beschreibung: Bei Angabe der Zeilenbandbreite wird der darin enthaltene Text gelistet. D.h., der List-Befehl ist der Standardbefehl, der gültig ist, wenn kein anderer Befehl eingegeben wird. Die aktuelle Zeile wird auf die letzte Zeile der Zeilenbandbreite gebracht. In dem besonderen Fall, in dem <Zeilenbandbreite> nicht angegeben wird (d.h. Null- Befehl), wird die nächste Zeile dieser Datei angezeigt (und wird zur neuen, z. Zt. gültigen Zeile).

Beispiele:

1. * Zeigt die gesamte Datei an; die derzeitig bearbeitete Zeile wird $
2. 1 Zeigt die 1. Zeile an und betrachtet diese als derzeitig zu bearbeitende Zeile
3. 1,. Listet alle Zeilen, angefangen von der 1. Zeile bis zur derzeitig bearbeiteten Zeile; die derzeitig bearbeitete Zeile wird nicht geändert.

7.4.4.13 Mount (Vorbereitung eines Diskettenlaufwerks)

Syntax: mouNT <Gerätename>

Beschreibung: Dieser Befehl bereitet den Zugriff auf das angegebene Gerät vor. Der Befehl sollte abgesetzt werden, wenn eine neue Diskette in ein Laufwerk der Disketteneinheit, Modell 2040, eingeführt wird.

7.4.4.14 Name

Syntax: nAME <Dateiname>

Beschreibung: Dieser Befehl setzt den zu benutzenden Dateinamen oder zeigt den derzeitig benutzten Dateinamen an, der durch Auslassung des entsprechenden Parameters beim GET- und PUT-Befehl vom System eingesetzt wird. Wird ein Dateiname nicht angegeben, dann wird der derzeitig benutzte Dateiname lediglich angezeigt. Bei Spezifikation eines Dateinamens wird dieser für zukünftige Operationen benutzt. Es ist hierbei zu bemerken, daß die Spezifikation eines Dateinamens im Zusammenhang mit dem PUT- oder GET-Befehl auch dazu führt, daß dieser als zukünftig zu benutzender Dateiname vom System angenommen wird.

7.4.4.15 Put (Schreiben)

Syntax: <Zeilenbandbreite> pUT <Dateiname>
<Zeilenbandbreite> pUT

Beschreibung: Durch den PUT-Befehl werden alle durch <Zeilenbandbreite> spezifizierten Zeilen auf die Datei gespeichert, die mit dem Namen <Dateiname> angegeben wird. Wird <Zeilenbandbreite> nicht angegeben, dann wird die gesamte Datei kopiert (gleich "*p <Dateiname>"). Wenn der Dateiname nicht spezifiziert wird, nimmt das System den derzeitig gültigen Dateinamen als Standardwert an. Bei Angabe eines Dateinamens wird dieser neu spezifizierte Dateiname zum derzeitig gültigen Dateinamen.

Beispiele: p demo Speicherung der gesamten Datei auf die Datei unter dem Namen 'demo'.

7.4.4.16 Rename (Umtaufen)

Syntax: renAME <Dateiname> <Dateiname>
renAME <Dateiname> to <Dateiname>

Beschreibung: Dieser Befehl ändert den Namen einer Datei, die mit dem ersten Dateinamen angesprochen wird in einen Namen, der durch den zweiten Dateinamen gegeben ist.

7.4.4.17 Run (Ausführung)

Syntax: run

Beschreibung: Wenn der Editor im Zusammenhang mit einer Programmsprache eingesetzt wird, führt dieser Befehl zur Ausführung der in der Datei gespeicherten Befehle. Durch die Betätigung der RUN-Taste wird das gleiche bewirkt.

7.4.4.18 Scratch (Löschen)

Syntax: scrATCH <Dateiname>

Beschreibung: Dieser Befehl führt zur Löschung der mit <Dateiname> angegebenen Datei.

7.4.4.19 Tabset (Tabulatorsetzung)

Syntax: tabSET <tab-1> <tab-2> <tab-3> ... <tab-n>
tabSET

Beschreibung: Dieser Befehl setzt neue Tabulatorstops oder zeigt die derzeitig gültigen an. Diese markieren die Positionen, auf die der Cursor springt, wenn die TAB-Taste gedrückt wird. Bei Angabe von Tabulatorstops werden neue Tabulatoren gesetzt. Ansonsten werden die derzeitig gültigen Tabulatorstops lediglich angezeigt. Es können bis zu 10 Tabulatorstops spezifiziert werden.

7.4.4.20 Talk

Syntax: tALK

Beschreibung: Das System wird in den 'Durchgangsmodus' versetzt und wird somit zu einem Terminal des Zentralrechners. Dieser Betriebsmodus wird mit der Betätigung der STOP-Taste beendet. Für eine erfolgreiche Kommunikation mit dem Zentralrechner sollte das HOST SETUP desselben vorher gestartet worden sein.

7.4.4.21 Time (Zeit)

Syntax: timE <Datum> <Zeit>
timE

Beschreibung: Dieser Befehl ist mit dem DATE-Befehl identisch, mit der Ausnahme, daß nur die Zeit angezeigt wird, wenn keine Parameter zusammen mit dem TIME-Befehl spezifiziert werden.

KAPITEL 8

KOMMUNIKATIONSPROGRAMM SETUP

8.1 EINLEITUNG

Der MMF 9000 ist mit einem RS232-Anschluß ausgerüstet. Dieser kann dazu verwendet werden, den MMF 9000 mit einem Zentralrechner zu verbinden. Nach dem Aufbau dieser Verbindung kann der MMF 9000 als Terminal zum Zentralrechner eingesetzt werden. Der TALK-Befehl des Waterloo MicroEditors versetzt den Benutzer in die Lage, mit dem Zentralrechner über die Tastatur Verbindung aufzunehmen. Ein im MMF 9000 laufendes Programm kann darüber hinaus dazu verwendet werden, unmittelbar mit dem Zentralrechner zu kommunizieren, indem Daten von dem Gerät 'serial' gelesen oder auf dieses geschrieben werden.

Andererseits kann der Zentralrechner als ein dem MMF 9000 angeschlossenes Gerät betrachtet werden. In diesem Fall muß zur Speicherung der Dateien innerhalb des Zentralrechner ein Kommunikationsprogramm laufen. Dieses steuert den Zugriff auf die Dateien des Zentralrechners. Das Programm wird nicht standardmäßig zusammen mit der Software zum Commodore MMF 9000 geliefert, kann jedoch für die verschiedenen Computersysteme von Waterloo angefordert werden. Der RS232-Anschluß kann auch dazu verwendet werden, ein Hilfsgerät (wie z.B. einen ASCII-Drucker) mit dem MMF 9000 zu verbinden.

Zentralrechner und andere Geräte kommunizieren unter Verwendung der seriell betriebenen Leitung mit unterschiedlichen Methoden. Zur Information des MMF 9000 über die Art und Weise in der der Zentralrechner die Kommunikation zwischen diesen beiden Geräten erwartet, kann das SETUP-Programm über das Hauptmenü aufgerufen werden.

Das SETUP-Programm zeigt auf dem Bildschirm verschiedene Kommunikationsoptionen an, zusammen mit deren Standardwerten. Diese Standardwerte können mit den entsprechend benötigten Informationen überschrieben werden. Dabei muß der Cursor entweder unter Verwendung der Cursor-Steuertasten oder der TAB-Tasten auf die entsprechende Stelle gebracht werden, wonach die Eingabe des gewünschten Wertes zu erfolgen hat. Nach der Abarbeitung aller Optionen ist die RETURN-Taste zu betätigen, wodurch der MMF 9000 sich auf die neuen Werte einstellt und das Hauptmenü angezeigt wird.

Wenn fehlerhafte Eingaben gemacht wurden, wird an der entsprechenden Stelle ein Fragezeichen (?) erscheinen und die Optionen werden nochmals angezeigt. An diese Stelle ist es möglich, die fehlerhaften Werte zu korrigieren, wonach die RETURN-Taste nochmals zu drücken ist.

Ein Benutzer sollte von dem für den Zentralrechner verantwortlichen Leiter die gültigen Werte erfragen, die als SETUP-Optionen einzusetzen sind. Die zur Verfügung gestellten Standardwerte beziehen sich auf VM/SP einer IBM-Anlage der Serie /370.

8.2 SETUP-OPTIONEN

Nach dem Laden des SETUP-Programms über das Hauptmenü werden die folgenden Informationen angezeigt:

Baud	2400	(Datenübertragungsrate)
Parity	EVEN	(Parität)
Stopbits	1	(Stop-Bits)
Prompt	11	(Abfrage)
Lineend	OD	(Zeilenende)
Response	13	(Antwort)

In diesem Abschnitt werden alle Optionen näher beschrieben und die gültigen Eingaben angezeigt. Beim Anschluß eines Hifsgeräts oder bei Benutzung des MMF 9000 als nicht-intelligentes Terminal sind nur die ersten drei Optionen von Bedeutung.

8.2.1 Baud-Rate (Datenübertragungsrate)

Die Datenübertragungsrate bestimmt die Geschwindigkeit, mit der Informationen über die seriell betriebene Leitung übertragen werden. Die folgenden Werte sind erlaubt:

50	75	110	135	150	300	600
1200	1800	2400	3600	4800	7200	9600

8.2.2 Parität

Die vier folgenden Einträge sind erlaubt: EVEN (gerade), ODD (ungerade), MARK, SPACE.

8.2.3 Stop-Bits

Es können entweder 1 oder 2 Stop-Bits gesendet werden, die sich zwischen den einzelnen Zeichen einer Informationskette befinden. Das bedeutet, daß eine gültige Antwort nur 1 oder 2 lauten kann.

8.2.4 Abfrage

Der Prompt-String ist eine Zeichensequenz, die vom Zentralrechner gesendet wird, um dem Mikroprozessor anzuzeigen, daß eine weitere Zeile von Eingabedaten gesendet werden kann. Der Prompt-String kann bis zu vier Zeichen lang sein und wird im hexadezimalen Format angegeben. Der Standardwert ist hexadezimal 11 (auch häufig XON-Zeichen genannt).

8.2.5 Zeilenende

Das Zeilenendezeichen wird dem Ende einer jeden Zeile hinzugefügt, welche an den Zentralrechner gesendet wird, um somit das Ende einer Zeile zu signalisieren. Der Wert besteht aus einem einzelnen Zeichen, welches im hexadezimalen Format eingegeben wird. Der Zentralrechner prüft jede einzelne, vom Mikrocomputer empfangene Zeile auf dieses Zeichen. Der Standardwert ist hexadezimal ØD (das ASCII-Wagenrücklaufzeichen; CR).

8.2.6 Antwort

Das Antwortzeichen wird immer dann vom Zentralrechner an den Mikrocomputer abgesendet, wenn der Zentralrechner eine Datenzeile empfangen hat. Der Wert dieser Option wird als einzelnes Zeichen im hexadezimalen Format angegeben. Dieses ist das letzte Zeichen einer vom Zentralrechner empfangenen Zeichenkette, als Antwort auf ein Zeilenendezeichen, welches vom Mikrokomputer abgesendet wurde. Der Standardwert ist hexadezimal 13 (häufig auch XOFF-Zeichen genannt).

ANHANG A

GLEITKOMMA-EMULATION

Die Hardware des Commodore MMF 9000 ist nicht in der Lage, Gleitkomma-arithmetik-Operationen auszuführen. Die Waterloo Microsoftware wurde so geschrieben, daß diese Operationen unter Verwendung der in der Bibiliothek gespeicherten Software ausgeführt werden kann. Dieser Anhang gibt einen Überblick über das Format dieser Werte.

Jeder Gleitkommawert wird unter Verwendung von 5 Bytes (40 Bits) gespeichert. Dabei ist das 1. Bit für das Vorzeichen dieser Zahl zuständig (1 = negativ). Die nächsten 8 Bits enthalten den Exponenten (zur Basis 2) im 2er-Komplement. Die Mantisse ist in den übrigen 31 Bits gespeichert und repräsentiert eine 32-Bit-Binärfraktion, in der das 1. Bit immer gleich Ø ist. Null wird als 40 Null-Bits dargestellt.

Das hier beschriebene Format erlaubt eine ungefähre Genauigkeit von 9 signifikanten Stellen. Wie weithin bekannt, werden sich eventuelle Fehler im gleichen Ausmaß erhöhen, je mehr Berechnungen unter Verwendung dieser Werte ausgeführt werden. Der Gesamtfehler ist jedoch im Vergleich zum Endergebnis bei den meisten Berechnungen unerheblich. Es würde den Rahmen dieses Handbuches sprengen, diese Probleme im Detail zu untersuchen.

Alle Kalkulationen werden intern unter Verwendung von 5-Byte-Mantissen ausgeführt. Dadurch bleibt der Fehler auch bei längeren Berechnungen klein.

ANHANG B

WATERLOO-BIBLIOTHEK

Wird der 6809-Mikroprozessor eingesetzt, dann ist es möglich, eine Unterprogramm-bibliothekt in Verbindung mit den verschiedenen Software-Programmen zu verwenden. Diese Routinen werden in dem Handbuch WATERLOO 6809 ASSEMBLER beschrieben. Die Routinen werden mit deren Namen angesprochen und befinden sich in der Datei 'WATLIB.EXP' auf der mitgelieferten Diskette.

ANHANG C

BILDSCHIRMHANDHABUNG

In diesem Anhang werden einige Bildschirmfunktionen beschrieben.

C.1 ZEICHENSÄTZE

Wie bereits vorher beschrieben, stehen vier unterschiedliche Zeichensätze zur Verfügung:

1. Commodore graphischer Zeichensatz,
2. Commodore Groß-/Kleinbuchstaben Roman-Zeichensatz,
3. Waterloo Groß-/Kleinbuchstaben Roman-Zeichensatz,
4. Waterloo APL-Zeichensatz.

Zu einem bestimmten Zeitpunkt wird einer dieser Zeichensätze für die Anzeige von Zeichen auf dem Bildschirm benutzt. Die Zeichendarstellung im Zusammenhang mit jedem einzelnen Zeichensatz kann angezeigt werden, indem ein kleines Programm erzeugt wird, welches die Zeichen der Werte zwischen Ø und 255 anzeigt. Diese Aufgabe könnte mit dem folgenden Waterloo MicroFORTRAN-Programm bewerkstelligt werden:

```
do  i = Ø, 255
    print, i, char(i)
enddo
end
```

Ähnliche Programme können unter Verwendung anderer Programmiersprachen entwickelt werden.

C.1.1 Zeichensatzselektion

Es stehen zwei Arten von Zeichensätze zur Verfügung: die Commodore-Zeichensätze und die Waterloo-Zeichensätze. Zur Selektion der Waterloo-Zeichensätze sollten die folgenden hexadezimalen Werte in die folgenden Adressen geschrieben werden:

Adresse	Wert
E880	ØC
E881	3Ø

Zur Selektion der Commodore-Zeichensätze sollten die folgenden hexadezimalen Werte in die folgenden Adressen geschreiben werden:

Adresse	Wert
E880	ØC
E881	1Ø

Diese Eintragungen in diese Positionen sind mittels der POKE-Befehle möglich, die in den verschiedenen Sprachen zur Verfügung stehen. Wurde ein entsprechendes Zeichensatzpaar ausgewählt, kann einer der zwei nun verfügbaren Zeichensätze selektiert werden, indem einer der folgenden hexadezimalen Werte in eine der folgenden Adressen eingegeben wird:

Adresse	Wert	
E84C	ØC	(Selektion des Zeichensatzes 1)
E84C	ØE	(Selektion des Zeichensatzes 2)

C.2 ENTFERNUNG ZUSÄTZLICHER BILDSCHIRMZEILEN

Wie bereits vorher beschrieben, wird jedes Zeichen unter Verwendung einer 8 x 8 Punktematrix angezeigt. Standardmäßig befinden sich zwei zusätzliche Punktereihen zwischen jeder Zeile. Diese Trennungszeilen können eliminiert werden, indem die folgenden hexadezimalen Werte in die folgenden Adresse (unter Beachtung der angegebenen Reihenfolge) geschrieben werden:

Adresse	**Wert**
E880	Ø4
E881	28
E880	Ø5
E881	Ø5
E880	Ø7
E881	21
E880	Ø9
E881	Ø7

Diese Überschreibung von Hauptspeicheradressen wird mittels der POKE-Befehle erreicht, die in den verschiedenen Sprachen zur Verfügung stehen. Sollten die zwei Trennungslinien auf dem Bildschirm wieder angezeigt werden, dann müssen die folgenden hexadezimalen Werte in die folgenden Adressen (in der beschriebenen Reihenfolge) eingegeben werden:

Adresse	**Wert**
E880	Ø4
E881	2Ø
E880	Ø5
E881	Ø3
E880	Ø7
E881	1D
E880	Ø9
E881	Ø9

Nachdruck, auch auszugsweise, nur mit schriftlicher Genehmigung von Commodore.



Commodore GmbH
Lyoner Straße 38
D-6000 Frankfurt/M. 71

Commodore AG
Aeschenvorstadt 57
CH-4010 Basel

Steiner Computer
Vertriebsgesellschaft mbH
Fleschgasse 2
A-1130 Wien

242a-482

Änderungen vorbehalten